Enhanced Differential-linear Cryptanalysis of Forró with MILP

Zhengting Li, Lin Ding, Xinhai Wang, Honglei Wang, Jiang Wan, and Fan Zhang

***Abstract*—ARX-based design is a major building block of modern cryptographic ciphers due to its efficiency in software. Forró is an ARX-based stream cipher proposed by Coutinho et al. at ASIACRYPT 2022, which was designed to provide higher security margin than the ChaCha stream cipher. In this paper, we propose a full automated MILP model called *MinForró*, to derive linear approximations for the Forró stream cipher. For the differential part, a two-stage strategy to search for single-bit differential trails with high differential correlations is presented, which helps us to find the first-ever 3-round differential trails for Forró. By combining the linear approximations obtained by *MinForró* and 3-round differential trail for Forró, we propose improved differential-linear distinguishers for 4-, 5-, 5.25-, 5.5-, 5.75-, 6-, 6.25- and 6.5-round Forró with complexities $2^{32.44}$, 2^{46}, 2^{50}, $2^{64.32}$, $2^{87.12}$, $2^{117.92}$, $2^{174.92}$ and $2^{226.88}$, respectively. The proposed differential-linear distinguishers for 4-, 5-, 5.25- and 5.5-round Forró significantly improve the existing distinguishers by factors of $2^{4.11}$, $2^{83.68}$, $2^{127.64}$ and $2^{178.20}$, respectively. To the best of our knowledge, this is the first differential-linear distinguisher for Forró that reaches 6.5 rounds, which is a significant advancement over the existing record of 5.5 rounds. We have implemented the differential-linear distinguishers for 4- and 5-round Forró on a common PC, and the experimental results confirm the correctness of these distinguishers. Furthermore, when combined with the *Probabilistic Neutral Bits* (PNB) technique, we obtain key recovery attacks on 5.5-, 6-, 6.5- and 6.75-round Forró with time complexities $2^{149.20}$, $2^{151.84}$, $2^{213.49}$ and $2^{251.97}$, respectively. The proposed key recovery attack on 5.5-round Forró significantly improves the time complexity of the existing attack by a factor of $2^{75.84}$. To the best of our knowledge, this is the first key recovery attack on Forró that reaches 6.75 rounds, which is a significant advancement over the existing record of 5.5 rounds.**

***Index Terms*—Differential-linear cryptanalysis, MILP, Probabilistic Neutral Bits, Forró, Stream cipher.**

I. Introduction

CRYPTOGRAPHY is an important tool for protecting information transmitted over networks. Among the most important cryptographic primitives, symmetric key primitives are an essential component. As a type of symmetric key primitives, stream ciphers are easy to implement and offer high encryption/decryption speeds. These advantages make them suitable for critical applications such as military and diplomatic communications, as well as mobile communication systems.

Z. Li, L. Ding, X. Wang, H. Wang and J. Wan are with the Information Engineering University, Zhengzhou, China, 450001. E-mail: d-inglin_cipher@163.com. (*Corresponding author: Lin Ding*.)

F. Zhang is with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China, 310058. E-mail: fanzhang@zju.edu.cn.

Manuscript received January 6, 2026; revised April 3, 2026; accepted May 10, 2026.

The ARX-based is a major building block of modern ciphers and is of immense interest in cryptography, due to its efficiency in software. ARX stands for **A**ddition (modulo a power of two), word-wise **R**otation and **X**OR. Due to the carry operation of modular addition, the algebraic degree of an ARX cipher is generally high after only a very few rounds, since the carry bit within one modular addition already reaches almost maximal degree. For the differential and linear attacks, the ARX-based ciphers generally possess strong resistance, as the differential probabilities and linear correlations decrease very quickly with an increasing number of rounds.

Salsa [1] is an ARX-based and widely-used stream cipher. It was designed by Bernstein in 2005 and selected into the final eSTREAM competition. In 2008, Berstein proposed the ChaCha stream cipher [2] as a variant of Salsa to provide better diffusion and higher resistance against cryptanalysis. Now, ChaCha has become one of the most widely used ciphers in the real world, such as in TLS, SSH and IPsec. In order to provide better diffusion per round and higher resistance to cryptanalysis, Coutinho et al. [3] made some modifications to ChaCha and proposed a new stream cipher called Forró at ASIACRYPT 2022. To achieve that, they introduced a new design strategy, called *Pollination*, constructed to speed up confusion and diffusion. They showed that Forró can achieve the same security as ChaCha with fewer operations, which allows them to reduce the total number of rounds while maintaining the security level. As a result, the designers recommended using Forró with a total of 14 rounds to achieve a security margin comparable with 20 rounds of ChaCha. Thus, Forró can achieve faster performance in many platforms (especially in constrained devices) than ChaCha. To enhance the authentication capability of Forró, Coutinho et al. [4] developed XForró14-Poly1305 in 2023, a new AEAD scheme that incorporates Poly1305.

A. Related Works

Up to now, there have been some cryptanalytic attacks on Forró published. In the specification of Forró [3] (and the extended version [5]), the designers of Forró tested all combinations of single-bit input difference (*ID*) and single-bit output difference (*OD*), and obtained four 2-round differential trails. They proposed differential-linear distinguishers for 3-, 4-, 5- and 5.25-round Forró with complexities $2^{18.9814}$, $2^{36.55}$, $2^{129.68}$ and $2^{176.81}$, respectively. They also proposed a key recovery attack on 5-round Forró with a time complexity of 2^{158}, by combining the *Probabilistic Neutral Bits* (PNB) technique. In [6], Sharma et al. enhanced the key recovery

TABLE I
DISTINGUISHERS AND KEY RECOVERY ATTACKS ON FORRÓ

Rounds	Attack Type	Time Complexity	Data Complexity	Ref.
3	Distinguisher	$2^{18.98}$	$2^{18.98}$	[3], [5]
4	Distinguisher	$2^{36.55}$	$2^{36.55}$	[3], [5]
	Distinguisher	$\mathbf{2^{32.44}}$	$\mathbf{2^{32.44}}$	**This paper**
5	Distinguisher	$2^{129.68}$	$2^{129.68}$	[3], [5]
	Distinguisher	$\mathbf{2^{46}}$	$\mathbf{2^{46}}$	**This paper**
	Key Recovery	2^{158}	2^{57}	[3], [5]
	Key Recovery	$2^{144.14}$	$2^{52.36}$	[6]
5.25	Distinguisher	$2^{179.56}$	$2^{179.56}$	[3], [5]
	Distinguisher	$2^{177.64}$	$2^{177.64}$	[7]
	Distinguisher	$\mathbf{2^{50}}$	$\mathbf{2^{50}}$	**This paper**
5.5	Distinguisher	2^{296}	2^{296}	[3], [5]
	Distinguisher	$2^{242.52}$	$2^{242.52}$	[7]
	Distinguisher	$\mathbf{2^{64.32}}$	$\mathbf{2^{64.32}}$	**This paper**
	Key Recovery	$2^{225.04}$	2^{47}	[6]
	Key Recovery	$\mathbf{2^{149.20}}$	$\mathbf{2^{40.37}}$	**This paper**
5.75	Distinguisher	$\mathbf{2^{87.12}}$	$\mathbf{2^{87.12}}$	**This paper**
6	Distinguisher	$\mathbf{2^{117.92}}$	$\mathbf{2^{117.92}}$	**This paper**
	Key Recovery	$\mathbf{2^{151.84}}$	$\mathbf{2^{45.65}}$	**This paper**
6.25	Distinguisher	$\mathbf{2^{174.92}}$	$\mathbf{2^{174.96}}$	**This paper**
6.5	Distinguisher	$\mathbf{2^{226.88}}$	$\mathbf{2^{226.88}}$	**This paper**
	Key Recovery	$\mathbf{2^{213.49}}$	$\mathbf{2^{57.45}}$	**This paper**
6.75	Key Recovery	$\mathbf{2^{251.97}}$	$\mathbf{2^{61.87}}$	**This paper**

attacks on Forró by introducing a multi-step approach and combining with the *syncopation* technique. They proposed key recovery attacks on 5- and 5.5-round Forró with time/data complexities $2^{144.14}$/$2^{52.36}$ and $2^{225.04}$/2^{47}, respectively. These are the best known key recovery attacks on Forró.

At ASIACRYPT 2025, Gong et al. [7] introduced a systematic re-decomposition technique for differential-linear distinguishers of ARX-based ciphers and applied it to Forró. They revisited the distinguisher for 5.25-round Forró proposed in [3] and [5], and corrected the complexity from $2^{176.81}$ to be $2^{179.56}$. More importantly, they proposed the first distinguisher for 5.5-round Forró with complexity $2^{242.52}$, which improves the cryptanalytic result (i.e., 2^{296}) given by the designers in [3] and [5] by a factor of about $2^{53.48}$.

B. Our Contributions

In this paper, we propose a fully automated MILP model to derive linear approximations for the Forró stream cipher, called *MinForró*, and a two-stage strategy to search for single-bit differential trails with high differential correlations. As applications, several differential-linear distinguishers and key recovery attacks on reduced-round Forró are presented, which achieve significant advancements over the existing records of attacked rounds. The existing differential-linear distinguishers and key recovery attacks on reduced-round Forró are listed in Table I. Our contributions are summarized as follows.

- ***Technical Contributions.*** In this paper, we propose a fully automated MILP model to derive linear approximations for the Forró stream cipher, called *MinForró*, which is carefully crafted to identify high-correlation linear trails. Utilizing *MinForró* with a single-bit input mask, the linear approximations for 1-, 2-, 2.25-, 2.5-, 2.75-, 3-, 3.25- and 3.5-round Forró that hold with correlations 1, 2^{-4}, 2^{-5}, 2^{-9}, 2^{-16}, 2^{-25}, 2^{-40} and 2^{-55} respectively are obtained. For the differential part, we propose a two-stage strategy to search for single-bit differential trails with high differential correlations. As a result, the first-ever 3-round differential trails for Forró are found, among which the best one achieves a differential correlation of $2^{-16.22}$.
- ***Distinguishers.*** By combining the linear approximations obtained by *MinForró* and 3-round differential trail for Forró, we propose improved differential-linear distinguishers for 4-, 5-, 5.25-, 5.5-, 5.75-, 6-, 6.25- and 6.5-round Forró with complexities $2^{32.44}$, 2^{46}, 2^{50}, $2^{64.32}$, $2^{87.12}$, $2^{117.92}$, $2^{174.92}$ and $2^{226.88}$, respectively. As shown in Table I, our differential-linear distinguishers for 4-, 5-, 5.25- and 5.5-round Forró significantly improve the existing distinguishers by factors of $2^{4.11}$, $2^{83.68}$, $2^{127.64}$ and $2^{178.20}$, respectively. To the best of our knowledge, this is the first differential-linear distinguisher for Forró that reaches 6.5 rounds, which is a significant advancement over the existing record of 5.5 rounds. We have

implemented the differential-linear distinguishers for 4- and 5-round Forró on a common PC, and the experimental results confirm the correctness of these distinguishers.

- ***Key Recovery Attacks.*** When combined with the PNB technique, we obtain key recovery attacks on 5.5-, 6-, 6.5- and 6.75-round Forró with time complexities $2^{149.20}$, $2^{151.84}$, $2^{213.49}$ and $2^{251.97}$, respectively. As shown in Table I, our key recovery attack on 5.5-round Forró significantly improves the time complexity of the existing attack by a factor of $2^{75.84}$. To the best of our knowledge, this is the first key recovery attack on Forró that reaches 6.75 rounds, which is a significant advancement over the existing record of 5.5 rounds.

As shown in Table I, it is clear to see that our cryptanalytic attacks on reduced-round Forró are much better than the existing attacks, which definitely demonstrates the superiority of the cryptanalytic techniques proposed in this paper.

C. Organization

The paper is organized as follows. In Section II, we provide a brief description of the Forró stream cipher, and introduce the basic ideas of differential-linear cryptanalysis and the PNB technique. In Section III, a fully automated MILP model to derive linear approximations for Forró, called *MinForró*, is proposed. In Section IV, we utilize *MinForró* to search for linear approximations of Forró. In Section V, a two-stage strategy to search for single-bit differential trails with high differential correlations is proposed, and then new differential-linear distinguishers for Forró are presented in Section VI. In Section VII, the PNB-based key recovery attacks on Forró are proposed, by combining with the PNB technique. Finally, Section VIII concludes the paper.

II. Preliminaries

This section provides a brief description of Forró, and introduces the basic ideas of differential-linear cryptanalysis and the PNB technique. To improve readability, we give some notations used throughout the paper in Table II.

TABLE II
SOME NOTATIONS

Notation	Description
X	The state matrix of the cipher (16 words)
$X^{(r)}$	State matrix after r rounds
$X^{[s]}$	State matrix after s subrounds
$x_i^{(r)}$	The i-th word of state $X^{(r)}$
$x_{i,j}^{(r)}$	The j-th bit of $x_i^{(r)}$
$x_i^{(r)}[j_0, j_1, \ldots, j_t]$	The sum $x_{i,j_0}^{(r)} \oplus x_{i,j_1}^{(r)} \oplus \cdots \oplus x_{i,j_t}^{(r)}$
$x + y$	Addition of x and y modulo 2^{32}
$x \oplus y$	Bitwise XOR of x and y
$x \lll l$	Rotation of x by l bits to the left

A. A Brief Description of Forró

At ASIACRYPT 2022, Coutinho et al. [5] proposed a new stream cipher called Forró in order to provides better diffusion per round and higher resistance to cryptanalysis. The cipher operates on 16 32-bit words, organized as a 4×4 matrix X. The state is initialized by a 256-bit key $k_0, \cdots, k_7$, a 64-bit nonce v_0, v_1, a 64-bit counter t_0, t_1 and 4 constants $c_0, \cdots, c_3$. We will refer to the nonce and counter words together as IV words. The initial state matrix of Forró is given as follows.

$$X^{(0)} = \begin{pmatrix} x_0^{(0)} & x_1^{(0)} & x_2^{(0)} & x_3^{(0)} \\ x_4^{(0)} & x_5^{(0)} & x_6^{(0)} & x_7^{(0)} \\ x_8^{(0)} & x_9^{(0)} & x_{10}^{(0)} & x_{11}^{(0)} \\ x_{12}^{(0)} & x_{13}^{(0)} & x_{14}^{(0)} & x_{15}^{(0)} \end{pmatrix} = \begin{pmatrix} k_0 & k_1 & k_2 & k_3 \\ t_0 & t_1 & c_0 & c_1 \\ k_4 & k_5 & k_6 & k_7 \\ v_0 & v_1 & c_2 & c_3 \end{pmatrix}$$

where $c_0 = 0x746C6F76$, $c_1 = 0x61616461$, $c_2 = 0x72626173$ and $c_3 = 0x61636E61$ are four constant words that are represent the little-endian encoding of the ASCII string "voltadaasabranca", a song by Brazilian artist. According to the designers, this choice is purely cultural, guaranteeing no compromise to the algorithm's security or performance.

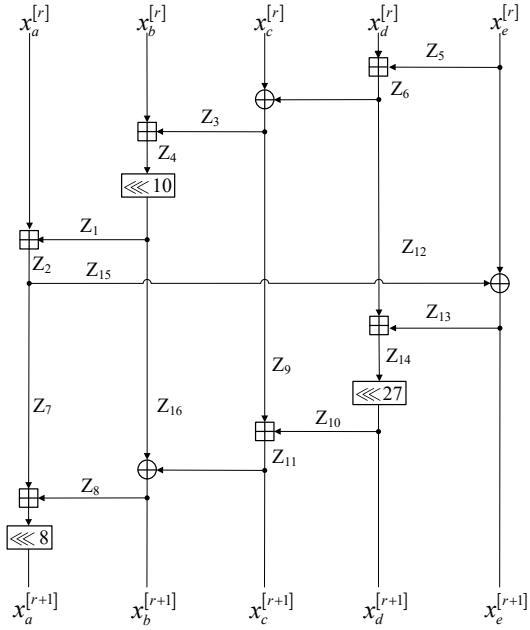


Fig. 1. The *SR* function of Forró with intermediate variables $Z_1, \ldots, Z_{16}$.

The *Round* function of Forró is remarkably different from that of ChaCha because of the *Pollination* technique. The designers of Forró chose this name as an analogy to the real Pollination in nature: when a bee collects nectar from a flower, the pollen sticks to the hairs of her body. The technique was proposed to maximize confusion and diffusion to bring non-linearity and confusion to other elements in the state matrix. The initial state matrix $X^{(0)}$ is updated by the *Round* function. Each *Round* function composes four *SubRound* (*SR*) function executed sequentially, i.e., $x^{(r)} = x^{[4r]}$. Based on the description provided by the designers,

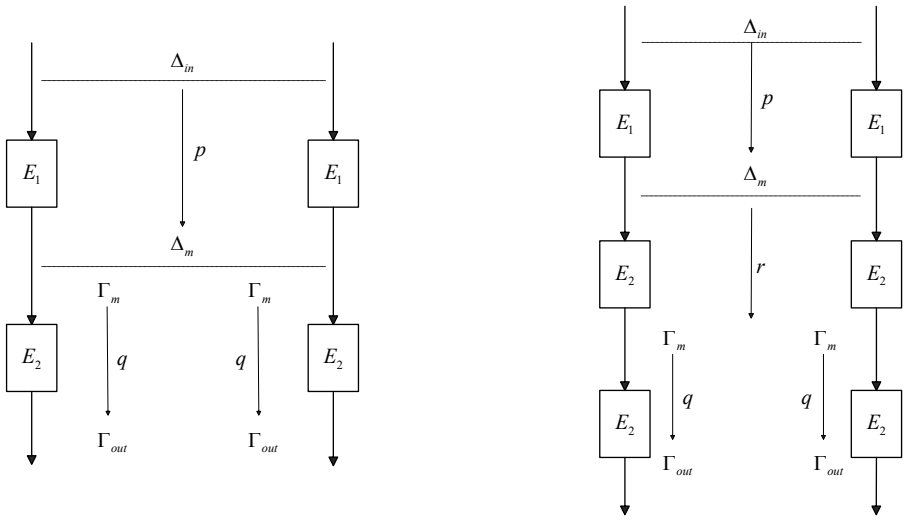


Fig. 2. A classical differential-linear distinguisher (on the left) and a differential-linear distinguisher with experimental evaluation of the correlation r.

we depict in Fig. 1 the *SR* function of Forró. Taking five words $\left(x_a^{[r]}, x_b^{[r]}, x_c^{[r]}, x_d^{[r]}, x_e^{[r]}\right)$ as input, we can obtain five output words $\left(x_a^{[r+1]}, x_b^{[r+1]}, x_c^{[r+1]}, x_d^{[r+1]}, x_e^{[r+1]}\right)$ by the *SR* function via four intermediate words:

$$\begin{aligned}
x_{d'}^{[r]} &= x_d^{[r]} + x_e^{[r]};\\
x_{c'}^{[r]} &= x_c^{[r]} \oplus x_{d'}^{[r]};\\
x_{b'}^{[r]} &= (x_b^{[r]} + x_{c'}^{[r]}) \lll 10;\\
x_{a'}^{[r]} &= x_a^{[r]} + x_{b'}^{[r]};\\
x_e^{[r+1]} &= x_e^{[r]} \oplus x_{a'}^{[r]};\\
x_d^{[r+1]} &= (x_{d'}^{[r]} + x_e^{[r+1]}) \lll 27;\\
x_c^{[r+1]} &= x_{c'}^{[r]} + x_d^{[r+1]};\\
x_b^{[r+1]} &= x_{b'}^{[r]} \oplus x_c^{[r+1]};\\
x_a^{[r+1]} &= (x_{a'}^{[r]} + x_b^{[r+1]}) \lll 8.
\end{aligned}$$

There is a difference between odd and even rounds. For odd rounds, the *SR* function operates on the four columns in the following order: $(x_0, x_4, x_8, x_{12}, x_3)$, $(x_1, x_5, x_9, x_{13}, x_0)$, $(x_2, x_6, x_{10}, x_{14}, x_1)$ and $(x_3, x_7, x_{11}, x_{15}, x_2)$. For even rounds, the *SR* function operates on the four diagonals in the following order: $(x_0, x_5, x_{10}, x_{15}, x_3)$, $(x_1, x_6, x_{11}, x_{12}, x_0)$, $(x_2, x_7, x_8, x_{13}, x_1)$ and $(x_3, x_4, x_9, x_{14}, x_2)$.

The keystream block Z after R rounds is obtained as $Z = X^{(0)} + X^{(R)}$, where $X^{(R)}$ denotes the state matrix after applying R consecutive *Round* functions to the initial state $X^{(0)}$. It should be noted that the *Round* function of Forró is reversible, i.e., the five input words of the *SR* function can be obtained by operating the reverse *SR* function on the five output words. The designers suggested using a total of 14 rounds of Forró to achieve a security margin comparable to 20-round ChaCha, while 10 rounds of Forró provide a margin exceeding that of 12-round ChaCha.

B. Differential-Linear Cryptanalysis

The differential-linear cryptanalysis was introduced by Langford and Hellman in [8]. Let E be a cipher and we can divide this cipher into two parts, i.e., $E = E_2 \circ E_1$, where E_1 and E_2 are sub ciphers, covering l and h rounds, respectively (see the left side of Fig. 2). In particular, suppose that the differential $\Delta_{in} \to \Delta_m$ holds with probability $p = \Pr_{x \in F_2^n}\left(E_1(x) \oplus E_1(x \oplus \Delta_{in}) = \Delta_m\right)$. Moreover, assume that the linear trail $\Gamma_m \to \Gamma_{out}$ holds with correlation $q = \mathbf{Cor}_{x \in F_2^n}\left[\langle \Gamma_m, x\rangle \oplus \langle \Gamma_{out}, E_2(x)\rangle\right]$. Under the assumption of independence between these two parts, we obtain a differential-linear distinguisher for $l + h$ rounds with correlation pq^2, leading to a time (and data) complexity of $p^{-2}q^{-4}$.

In practice, correlation estimates can be inaccurate due to the assumption of independence between E_1 and E_2. Thus, to mitigate this issue, a common strategy is to divide the cipher into three parts, i.e., $E = E_2 \circ E_m \circ E_1$ (see the right side of Fig. 2), by adding a middle layer E_m. The middle part E_m is experimentally evaluated. In particular, suppose that $\Delta_m \overset{E_m}{\to} \Gamma_m$ holds with correlation $r = \mathbf{Cor}_{x \in S}\left[\langle \Gamma_m, E_m(x)\rangle \oplus \langle \Gamma_m, E_m(x \oplus \Delta_m)\rangle\right]$, where S is the set of samples over which the correlation r is computed. Then the total correlation can be estimated as prq^2, and thus the complexity of the differential-linear distinguisher is $p^{-2}r^{-2}q^{-4}$.

C. Probabilistic Neutral Bits

At FSE 2008, Aumasson et al. [9] firstly introduced the PNB technique and applied it to ChaCha. Before introducing the PNB technique, we first give a brief description of a concept *Neutrality Measure*. The neutrality measure of the key bit k_i is defined as γ_i, which refers to the correlation observed when flipping the key bit k_i does not affect the output keystream. In the PNB-based attack, the attacker sets

a threshold γ and computes the neutrality measure for each key bit γ_i. The key bits with $\gamma_i \leq \gamma$ are called *significant key bits*, indicating their substantial influence on the position where the differential is observed. This technique can divide the key bits into two categories: significant key bits (i.e., non-PNBs) and non-significant key bits (i.e., PNBs). Unlike a brute-force attack that requires guessing all key bits, only the significant key bits should be guessed to calculate the backward correlation in this technique. The remaining PNBs can be recovered through an exhaustive search.

Assume the differential Δ_{in} is fixed, and the attacker can obtain both the keystream Z (without the difference) and Z' (with the difference) of R rounds for nonce v, counter t and unknown key k. Then the attacker guesses the key k and inverts h rounds of the algorithm to get $X^{(l)}$ and $X'^{(l)}$ and compute $\Delta X^{(l)}$, if $\Delta X^{(l)} = \Delta_m$, the guessed key is correct key. Let f denote the function representing the aforementioned process, i.e., $f\left(k, v, t, Z, Z'\right) = \Delta_m$. There is $f\left(\hat{k}, v, t, Z, Z'\right) = \Delta_m$ with correlation ε_d if $\hat{k} = k$. Let $\bar{k}$ be a subkey of m significant key bits extracted from k. There is a function g with $\bar{k}$ as input and let f to be correlated to g with correlation ε_a, i.e., $\Pr\left(f\left(k, v, t, Z, Z'\right) = g\left(\bar{k}, v, t, Z, Z'\right)\right) = \frac{1}{2}\left(1 + \varepsilon_a\right)$. Denote the correlation of g by ε, under a reasonable independence assumption, $\varepsilon = \varepsilon_d \cdot \varepsilon_a$ holds.

Using **Neyman-Pearson Lemma**, the number of samples N required to get the bounds on probability of false alarm $p_{fa} = 2^{-\alpha}$ and non-detection $p_{nd} = 1.3 \times 10^{-3}$ is

$$N = \left(\frac{\sqrt{\alpha \log 4} + 3\sqrt{1 - \varepsilon_d^2 \varepsilon_a^2}}{\varepsilon_d \varepsilon_a}\right)^2,$$

and the time complexity is

$$2^{256-n} N + 2^{256-\alpha},$$

where α is a parameter that the attacker can choose and n is the number of PNBs.

At EUROCRYPT 2022, Dey et al. [10] provided a modified expression for the time complexity to achieve a more accurate measurement as $2^{256-n}N + 2^{256-\alpha} + 2^n$. Besides these works, there are many PNB-based key recovery attacks that have been published, for further information we refer to [**?**], [3], [9]–[16], [18]–[26].

A Systematic Three-Step Strategy to Find a Good Set of PNBs. Finding the PNBs is a critical step of proposing a valid PNB-based key recovery attack. To find a good set of PNBs, Dey et al. [10] proposed a systematic three-step strategy which yields a better PNB set compared with the conventional method described in [9]. The proposed strategy is briefly described as follows.

Step 1: ***Preliminary shortlisting and direct inclusion***

- In this step, the attacker assigns a threshold γ_{prelim} and selects the ones which give a higher bias than the threshold. Assume that n_{prelim} candidates are selected in this step. By assigning another threshold γ_{direct} which is higher than γ_{prelim}, the attacker includes the key bits with higher biases than γ_{direct} in the PNB set. Assume that n_1 PNBs are selected in this step.

Step 2: ***Selection of the best candidate in each iteration***

- In this step, the attacker includes some more PNBs (say n_2) by repeating the following iteration n_2 times. In the i-th iteration, the n_1 PNBs found in Step 1 and $(i-1)$ PNBs obtained in this step are assigned random values, and then the attacker changes the value one at a time and observes the backward bias among the remaining $(n_{prelim} - n_1)$ key bits. At the end of this step, a total of $(n_1 + n_2)$ PNBs can be obtained.

Step 3: ***Cherry-pick from the remaining while randomising the selected ones***

- The attacker assigns random values to the (n_1+n_2) PNBs and randomly changes one of the remaining shortlisted $(n_{prelim} - n_1 - n_2)$ candidates. Then the attacker calculates the backward bias and arranges all the candidates in descending order of their biases. Finally, the attacker selects the required number of PNBs from the top directly.

III. *MinForró*: A Fully Automated MILP Model to Derive Linear Approximations for Forró

Nowadays, automated tools for finding differential characteristics and linear approximations have attracted significant research attentions as they can rapidly discover high-probability differentials and high-correlation linear approximations and reduce the possibility of mistakes when the modeling is not too complex. In this section, we use constraints for characterizing the three operations of ARX ciphers. Using linear inequalities to model the components of Forró, we ultimately design a fully automated MILP model to derive linear approximations for Forró, called *MinForró*.

Constraints of the XOR Operation. The mask propagation through a XOR operation, with input masks $\alpha, \beta \in \mathbb{F}_2^n$ and output mask $\gamma \in \mathbb{F}_2^n$, is defined by the bit-wise equality:

$$\alpha_j = \beta_j = \gamma_j, \quad \forall j \in \{0, \ldots, n-1\}.$$

Constraints of the Three-Forked Branch. For any three-forked branch operation with input masks $\alpha, \beta \in \mathbb{F}_2^n$ and output mask $\gamma \in \mathbb{F}_2^n$, the constraints at bit level for each $j \in \{0, \ldots, n-1\}$ are defined by the following inequalities:

$$\begin{aligned} d_{\lambda_j} &\geq \alpha_j \\ d_{\lambda_j} &\geq \beta_j \\ d_{\lambda_j} &\geq \gamma_j \\ \alpha_j + \beta_j + \gamma_j &\geq 2d_{\lambda_j} \\ \alpha_j + \beta_j + \gamma_j &\leq 2 \end{aligned} \tag{1}$$

where d_{λ_j} is a dummy (auxiliary) variable.

Constraints of the Modular Addition. For any modular addition operation with input masks $\alpha, \beta \in \mathbb{F}_2^n$ and output mask $\gamma \in \mathbb{F}_2^n$, the constraints at bit level for each $i \in \{0, \ldots, n-1\}$ are defined by the following inequalities:

$$
\begin{aligned}
s_i - \gamma_i - \alpha_i + \beta_i + s_{i+1} &\geq 0 \\
s_i + \gamma_i + \alpha_i - \beta_i - s_{i+1} &\geq 0 \\
s_i + \gamma_i - \alpha_i - \beta_i + s_{i+1} &\geq 0 \\
s_i - \gamma_i + \alpha_i - \beta_i + s_{i+1} &\geq 0 \\
s_i + \gamma_i - \alpha_i + \beta_i - s_{i+1} &\geq 0 \\
s_i - \gamma_i + \alpha_i + \beta_i - s_{i+1} &\geq 0 \\
\gamma_i - s_i + \alpha_i + \beta_i + s_{i+1} &\geq 0 \\
s_i + \gamma_i + \alpha_i + \beta_i + s_{i+1} &\leq 4 \\
s_{n-1} &= 0
\end{aligned} \tag{2}
$$

where s_i is the probability weight variable.

Constraints for *SubRound*. In the *SR*, auxiliary variables Z_i for $1 \leq i \leq 16$ are created. Fig. 1 illustrates the new auxiliary variables. The modular addition constraints from Equation (2) apply to the following six operations, with their respective masks (α, β, γ) defined as:

$$
\begin{aligned}
(\alpha, \beta, \gamma) \in \big\{ & (x_a^{[r]}, Z_1, Z_2),\ (x_b^{[r]}, Z_3, Z_4), \\
& (x_d^{[r]}, Z_5, Z_6),\ (Z_7, Z_8, x_b^{[r+1]} \lll 8), \\
& (Z_9, Z_{10}, Z_{11}),\ (Z_{12}, Z_{13}, Z_{14}) \big\}
\end{aligned} \tag{3}
$$

The three-forked branch constraints from Equation (1) apply to the following nine operations, with their respective masks (α, β, γ) defined as:

$$
\begin{aligned}
(\alpha, \beta, \gamma) \in \big\{ & (Z_2, Z_7, Z_{15}),\ (Z_4 \lll 10, Z_1, Z_{16}), \\
& (x_c^{[r]}, Z_3, Z_9),\ (Z_6, x_c^{[r]}, Z_{12}), \\
& (x_e^{[r]}, Z_5, Z_{15}),\ (Z_{16}, Z_8, x_b^{[r+1]}), \\
& (Z_{11}, Z_{16}, x_c^{[r+1]}),\ (Z_{15}, Z_{13}, x_e^{[r+1]}), \\
& (Z_{14} \lll 27, Z_{10}, x_d^{[r+1]}) \big\}
\end{aligned} \tag{4}
$$

Each *Round* function composes four *SR* function, executed sequentially. This structure necessitates modeling Forró at the sub-round level rather than at the full-round level. Consequently, for Forró with a total of R *SR* functions, our model need $5 \times n \times (R+1)$ variables (where $n = 32$) to model the input and output masks of each *SR* function; we need $36 \times n \times R$ variables for modeling the dummy variable d_λ while using in the three-forked branch operation; we need $6 \times n \times R$ variables for modeling the probability weight variable s while using in the modular addition operation; we need $16 \times R$ variables for modeling the auxiliary variable Z. In total, our model comprises $47nR+16R+5n$ variables. The inequalities are distributed across two operations: $9 \times 6 \times n \times R$ from modular addition and $5 \times 9 \times n \times R$ from three-forked branches. Consequently, the total number of inequalities is $99nR$.

Objective Function. Given R rounds of *SR* to model and the probability weight variables $s_{i,j}^r$ (for bit j of the i-th modular addition in *SubRound* r), the optimization problem aims to minimize the total sum:

$$
\sum_{r=1}^{R} \sum_{i=0}^{6} \sum_{j=0}^{n-1} s_{i,j}^r.
$$

Differences from the Existing MILP Models. At FSE 2016, Fu et al. [27] proposed an MILP-based method for automatic search for differential characteristics and linear approximations in ARX-based block ciphers, and applied their method to the lightweight block cipher Speck. In IACR ToSC 2023, Bellini et al. [21] adapted the automatic methodology to construct a MILP model for the ChaCha stream cipher, and proposed several distinguishers and key recovery attacks on ChaCha256.

To highlight the innovation and importance of *MinForró*, we explain the reasons of proposing an automatic model for Forró from the following two aspects. On one hand, the Forró was designed at ASIACRYPT 2022, which aims at providing better diffusion per round and higher resistance to cryptanalysis with fewer operations than ChaCha. Although Forró has been analyzed by the designers and several cryptanalysts, no automated MILP tool is proposed for the cipher up to now. Thus, the security margin of Forró against the automated cryptanalytic tools is not clear. On the other hand, the existing MILP model for ChaCha cannot be directly migrated to Forró due to the fundamental differences in their structure and operations. This incompatibility arises from two principal divergences in the cipher design:

- **Structure Difference.** The state matrices of both ChaCha and Forró are updated by the *Round* function, which composes four sub-round functions (termed *QR* for ChaCha and *SR* for Forró). The fundamental architectural distinction lies in their execution patterns: the *QR* function is applied in parallel for ChaCha but the *SR* function is applied in a sequential order for Forró. The shift from a parallel to a sequential schedule represents a critical structural divergence. This structural difference makes the round-level constraints proposed in the earlier work inapplicable to Forró.
- **Operational Difference.** Forró uses more modular addition operations to replace some specific word-wise rotations. In the MILP model, the objective function is formulated as a sum of probability weight variables, which are inherently tied to the modular addition operations. Therefore, any variation in the modular addition operations will directly change the objective function. As a result, the objective function developed for ChaCha is not suitable for Forró.

Considering the two aspects above, we propose an automated MILP model to derive linear approximations for Forró called *MinForró*. The key difference of our model from the existing MILP models lies in the level of modeling. The existing MILP models characterize the full *Round* function as an indivisible unit, while we model each *SR* function individually. This difference leads to a substantial reduction in the number of constraints. More specifically, for an R-round instance of Forró, our MILP model requires only $4 \cdot 99nR$ inequalities, compared with the $800nR - 144R$ inequalities required for a R-round instance of ChaCha. By significantly reducing the number of constraints, *MinForró* achieves markedly higher computational efficiency while maintaining the analytical precision.

TABLE III
THE CORRELATION OF LINEAR APPROXIMATIONS FOR THE 16 INPUT BITS

Starting Bit	Correlation (3-6 Rounds)	Running Time	Correlation (3-6.5 Rounds)	Running Time
$x_{0,0}^{(3)}$	2^{-63}	3.12m	-	>24h
$x_{1,0}^{(3)}$	2^{-57}	3.13m	-	>24h
$x_{2,0}^{(3)}$	2^{-44}	1.07m	-	>24h
$x_{3,0}^{(3)}$	2^{-71}	9.88m	-	>24 h
$x_{4,0}^{(3)}$	2^{-41}	30s	2^{-85}	28.10m
$x_{5,0}^{(3)}$	2^{-35}	19s	2^{-65}	7.27m
$x_{6,0}^{(3)}$	2^{-50}	30s	2^{-92}	1.23h
$x_{7,0}^{(3)}$	2^{-49}	44s	2^{-91}	1.43h
$x_{8,0}^{(3)}$	2^{-33}	43s	2^{-66}	8.03h
$\boldsymbol{x_{9,0}^{(3)}}$	$\mathbf{2^{-25}}$	**31s**	$\mathbf{2^{-55}}$	**39.50m**
$x_{10,0}^{(3)}$	2^{-44}	1.08m	2^{-80}	15.38h
$x_{11,0}^{(3)}$	2^{-34}	1.13m	2^{-65}	1.50h
$x_{12,0}^{(3)}$	2^{-60}	9.85m	-	>24h
$x_{13,0}^{(3)}$	2^{-50}	46s	-	>24h
$x_{14,0}^{(3)}$	2^{-33}	16s	2^{-65}	39.93m
$x_{15,0}^{(3)}$	2^{-78}	2.68h	-	>24h

IV. LINEAR APPROXIMATIONS FOR FORRÓ WITH MILP

In this section, we utilize the MILP model *MinForró* to derive linear approximations for Forró with high correlations. Since the ARX-based ciphers like Forró are designed to providing good diffusion, the linear trails with few active input bits are often better than that with more active input bits. Thus, we will consider the linear trails for Forró with one active input bit, in a hope to find good linear trails for the cipher. Since the state matrix of Forró consists of 16 words in total, thus the number of the linear trails that should be considered is $16 \times 32 = 512$. Moreover, the active input bit appearing in the least significant bit (LSB) often performs better than that appearing in any other bit. Thus, we have successfully limited the number of the linear trails that should be considered to be 16, which reduces the search space for good linear trails efficiently.

To find the good linear approximations for Forró, we test all these 16 scenarios with one active input bit by *MinForró*. All linear trails start at the third round, as a 3-round differential trail for Forró has been proposed and will be presented in the following section. The results obtained are summarized in Table III. Take the tenth row as an example. The starting bit $x_{9,0}^{(3)}$ denotes the unique active bit at the third round. 2^{-25} denotes the correlation of the 3-round linear approximation obtained by *MinForró*, starting at the third round and ending at the sixth round, and the 3-round linear approximation is obtained with 31 seconds on a common PC. Similarly, 2^{-55} denotes the correlation of the 3.5-round linear approximation obtained by *MinForró*, starting at the third round, and the 3.5-round linear approximation is obtained with 39.50 minutes on a common PC. As shown in Table III, for some starting bits, the running times to derive 3.5-round linear approximations are much bigger than 24 hours and no optimal (highest-correlation) results are outputted by *MinForró* within a tolerable time frame, we denote them by the symbol '-'.

The Table III clearly shows that the starting bit $x_{9,0}^{(3)}$ is the best choice among all 16 scenarios, as it achieves the highest linear correlations of the 3- and 3.5-round linear approximations. It is important to note that we also tried the linear trails with more active input bits and the linear trails with the active input bit appearing in the non-least significant bit. As results, these linear trails do not perform better than the 1-bit input differences.

Now, we will list the linear approximations obtained by *MinForró*, with $x_{9,0}^{(3)}$ as the unique active bit at the third round. Recall the description of Forró, each *Round* function composes four *SubRound* (*SR*) function executed sequentially, i.e., $x^{(r)} = x^{[4r]}$.

Linear Approximation 1. The following 1-round linear approximation for Forró holds with probability 1:

$$x_9^{[12]}[0] = x_2^{[16]}[0] \oplus x_9^{[16]}[0] \oplus x_{14}^{[16]}[0, 27].$$

Computational Result 1. *The **Linear Approximation 1** holds computationally with $\varepsilon_0 = 1$. This correlation was verified using 2^{24} random samples.*

Linear Approximation 2. The following 2-round linear approximation for Forró holds with probability $\frac{1}{2}(1 + \frac{1}{2^4})$:

$$\begin{aligned} x_9^{[12]}[0] = {} & x_0^{[20]}[0] \oplus x_2^{[20]}[2, 3] \oplus x_3^{[20]}[10, 11] \\ & x_6^{[20]}[0, 26, 27] \oplus x_7^{[20]}[2, 3] \oplus x_9^{[20]}[0] \\ & x_{10}^{[20]}[0] \oplus x_{13}^{[20]}[0, 27] \oplus x_{14}^{[20]}[22, 27]. \end{aligned}$$

Computational Result 2. *The **Linear Approximation 2** holds computationally with $\varepsilon_1 = 2^{-3.39}$. This correlation was verified using 2^{30} random samples.*

Linear Approximation 3. The following 2.25-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^5})$:

$$\begin{aligned} x_9^{[12]}[0] = {} & x_0^{[21]}[8,18,19] \oplus x_2^{[21]}[2,3] \oplus \\ & x_3^{[21]}[0,10,11] \oplus x_5^{[21]}[10,11] \oplus \\ & x_6^{[21]}[0,26,27] \oplus x_7^{[21]}[2,3] \oplus x_9^{[21]}[0] \oplus \\ & x_{13}^{[21]}[0,27] \oplus x_{14}^{[21]}[22,27] \oplus x_{15}^{[21]}[0,27]. \end{aligned}$$

Computational Result 3. *The **Linear Approximation 3** holds computationally with $\varepsilon_2 = 2^{-4.39}$. This correlation was verified using 2^{30} random samples.*

Linear Approximation 4. The following 2.5-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^9})$:

$$\begin{aligned} x_9^{[12]}[0] = {} & x_0^{[22]}[8,18,19] \oplus x_1^{[22]}[16,26,27] \\ & x_2^{[22]}[2,3] \oplus x_3^{[22]}[0,10,11] \oplus \\ & x_5^{[22]}[10,11] \oplus x_6^{[22]}[4,5,7,8,10,18,19] \oplus \\ & x_7^{[22]}[2,3] \oplus x_9^{[22]}[0] \oplus x_{11}^{[22]}[0,4,5,10,26,27] \oplus \\ & x_{12}^{[22]}[0,26,27] \oplus x_{13}^{[22]}[0,27] \oplus \\ & x_{14}^{[22]}[22,27] \oplus x_{15}^{[22]}[0,27]. \end{aligned}$$

Computational Result 4. *The **Linear Approximation 4** holds computationally with $\varepsilon_3 = 2^{-7.97}$. This correlation was verified using 2^{32} random samples.*

Linear Approximation 5. The following 2.75-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^{16}})$:

$$\begin{aligned} x_9^{[12]}[0] = {} & x_0^{[23]}[8,18,19] \oplus x_1^{[23]}[16,26,27] \oplus \\ & x_2^{[23]}[8,10,11,24] \oplus x_3^{[23]}[0,10,11] \oplus \\ & x_5^{[23]}[10,11] \oplus x_6^{[23]}[4,5,7,8,10,18,19] \oplus \\ & x_7^{[23]}[0,12,13,15,16] \oplus x_8^{[23]}[12,13] \oplus \\ & x_9^{[23]}[0] \oplus x_{14}^{[23]}[22,27] \oplus \\ & x_{11}^{[23]}[0,4,5,10,26,27] \oplus \\ & x_{12}^{[23]}[0,26,27] \oplus x_{13}^{[23]}[2,3,22,27] \oplus \\ & x_{15}^{[23]}[0,27]. \end{aligned}$$

Computational Result 5. *The **Linear Approximation 5** holds computationally with $\varepsilon_4 = 2^{-13.67}$. This correlation was verified using 2^{36} random samples.*

Besides the results above, we have also obtained a 3-round linear approximation for Forró by *MinForró*. Before introducing it, a 0.25-round linear approximation for Forró is proposed as follows, as it will be used to calculate the computational correlation of the 3-round linear approximation for Forró.

Linear Approximation 6. The following 0.25-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^9})$:

$$\begin{aligned} & x_0^{[23]}[8,18,19] \oplus x_1^{[23]}[16,26,27] \oplus x_2^{[23]}[8,10,11,24] \oplus \\ & x_3^{[23]}[0,10,11] \oplus x_5^{[23]}[10,11] \oplus x_7^{[23]}[0,12,13,15,16] \oplus \\ & x_6^{[23]}[4,5,7,8,10,18,19] \oplus x_8^{[23]}[12,13] \oplus x_9^{[23]}[0] \oplus \\ & x_{14}^{[23]}[22,27] \oplus x_{11}^{[23]}[0,4,5,10,26,27] \oplus x_{12}^{[23]}[0,26,27] \oplus \\ & x_{13}^{[23]}[2,3,22,27] \oplus x_{15}^{[23]}[0,27] \\ = {} & x_0^{[24]}[8,18,19] \oplus x_1^{[24]}[16,26,27] \oplus x_2^{[24]}[0,8,10,11,24] \oplus \\ & x_3^{[24]}[0,2,3,8,16,29,30] \oplus x_5^{[24]}[10,11] \oplus \\ & x_6^{[24]}[4,5,7,8,10,18,19] \oplus x_7^{[24]}[0,12,13,15,16] \oplus \\ & x_4^{[24]}[7,8,10,11,21,22,23,24,26,27] \oplus x_8^{[24]}[12,13] \oplus \\ & x_9^{[24]}[10,11] \oplus x_{11}^{[24]}[0,4,5,10,26,27] \oplus x_{12}^{[24]}[0,26,27] \oplus \\ & x_{13}^{[24]}[2,3,22,27] \oplus x_{14}^{[24]}[0,17,22,27] \oplus x_{15}^{[24]}[0,27]. \end{aligned}$$

Computational Result 6. *The **Linear Approximation 6** holds computationally with $\varepsilon_5 = 2^{-7.70}$. This correlation was verified using 2^{32} random samples.*

By the **Piling-up Lemma**, we combine **Linear Approximation 5** and **Linear Approximation 6** to obtain the following 3-round linear approximation for Forró.

Linear Approximation 7. The following 3-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^{25}})$:

$$\begin{aligned} x_9^{[12]}[0] = {} & x_0^{[24]}[8,18,19] \oplus x_2^{[24]}[0,8,10,11,24] \oplus \\ & x_8^{[24]}[12,13] \oplus x_3^{[24]}[0,2,3,8,16,29,30] \oplus \\ & x_4^{[24]}[7,8,10,11,21,22,23,24,26,27] \oplus \\ & x_6^{[24]}[4,5,7,8,10,18,19] \oplus x_9^{[24]}[10,11] \oplus \\ & x_7^{[24]}[0,12,13,15,16] \oplus x_{11}^{[24]}[0,4,5,10,26,27] \oplus \\ & x_{12}^{[24]}[0,26,27] \oplus x_1^{[24]}[16,26,27] \oplus \\ & x_5^{[24]}[10,11] \oplus x_{14}^{[24]}[0,17,22,27] \oplus \\ & x_{13}^{[24]}[2,3,22,27] \oplus x_{15}^{[24]}[0,27]. \end{aligned}$$

Computational Result 7. *Using the **Piling-up Lemma** and **Computational Results 5 and 6**, the linear approximation of **Linear Approximation 7** holds computationally with $\varepsilon_6 = \varepsilon_4\varepsilon_5 = 2^{-(7.70+13.67)} = 2^{-21.37}$.*

We have also obtained a 3.25-round linear approximation for Forró by *MinForró*. To form this linear approximation, four linear approximations are proposed, i.e., a 2.5-round linear approximation and three 0.25-round linear approximations. They are described as follows.

Linear Approximation 8. The following 2.5-round linear

approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^9})$:

$$\begin{aligned} x_9^{[12]}[0] = {} & x_0^{[22]}[8,19] \oplus x_1^{[22]}[16,27]\, x_2^{[22]}[3] \oplus \\ & x_3^{[22]}[0,11] \oplus x_5^{[22]}[10,11] \oplus \\ & x_6^{[22]}[4,5,7,8,10,18,19] \oplus x_7^{[22]}[2,3] \oplus \\ & x_9^{[22]}[0] \oplus x_{11}^{[22]}[0,4,5,10,26,27] \oplus \\ & x_{12}^{[22]}[0,26,27] \oplus x_{13}^{[22]}[0,27] \oplus \\ & x_{14}^{[22]}[22,27] \oplus x_{15}^{[22]}[0,27]. \end{aligned}$$

Computational Result 8. *The **Linear Approximation 8** holds computationally with $\varepsilon_7 = 2^{-7.96}$. This correlation was verified using 2^{32} random samples.*

Linear Approximation 9. The following 0.25-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^8})$:

$$\begin{aligned} & x_0^{[22]}[8,19] \oplus x_1^{[22]}[16,27]\, x_2^{[22]}[3] \oplus x_3^{[22]}[0,11] \oplus \\ & x_5^{[22]}[10,11] \oplus x_6^{[22]}[4,5,7,8,10,18,19] \oplus x_7^{[22]}[2,3] \oplus \\ & x_9^{[22]}[0] \oplus x_{11}^{[22]}[0,4,5,10,26,27] \oplus x_{12}^{[22]}[0,26,27] \oplus \\ & x_{13}^{[22]}[0,27] \oplus x_{14}^{[22]}[22,27] \oplus x_{15}^{[22]}[0,27] \\ = {} & x_0^{[23]}[8,19] \oplus x_1^{[23]}[16,27] \oplus x_2^{[23]}[8,11,22,24] \oplus \\ & x_3^{[23]}[0,11] \oplus x_5^{[23]}[10,11] \oplus x_6^{[23]}[4,5,7,8,10,18,19] \oplus \\ & x_7^{[23]}[0,13,16] \oplus x_8^{[23]}[13] \oplus x_9^{[23]}[0] \oplus \\ & x_{11}^{[23]}[0,4,5,10,26,27] \oplus x_{12}^{[23]}[0,26,27] \oplus \\ & x_{13}^{[23]}[3,22,27] \oplus x_{14}^{[23]}[22,27] \oplus x_{15}^{[23]}[0,27]. \end{aligned}$$

Computational Result 9. *The **Linear Approximation 9** holds computationally with $\varepsilon_8 = 2^{-6.70}$. This correlation was verified using 2^{32} random samples.*

Linear Approximation 10. The following 0.25-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^{10}})$:

$$\begin{aligned} & x_0^{[23]}[8,19] \oplus x_1^{[23]}[16,27] \oplus x_2^{[23]}[8,11,22,24] \oplus \\ & x_3^{[23]}[0,11] \oplus x_5^{[23]}[10,11] \oplus x_6^{[23]}[4,5,7,8,10,18,19] \oplus \\ & x_7^{[23]}[0,13,16] \oplus x_8^{[23]}[13] \oplus x_9^{[23]}[0] \oplus \\ & x_{11}^{[23]}[0,4,5,10,26,27] \oplus x_{12}^{[23]}[0,26,27] \oplus \\ & x_{13}^{[23]}[3,22,27] \oplus x_{14}^{[23]}[22,27] \oplus x_{15}^{[23]}[0,27] \\ = {} & x_0^{[24]}[8,19] \oplus x_1^{[24]}[16,27] \oplus x_2^{[24]}[0,8,11,22] \oplus \\ & x_3^{[24]}[2,3,8] \oplus x_4^{[24]}[11,27] \oplus x_5^{[24]}[10,11] \oplus \\ & x_6^{[24]}[4,5,7,8,10,18,19] \oplus x_7^{[24]}[0,13,16] \oplus x_8^{[24]}[13] \oplus \\ & x_9^{[24]}[11] \oplus x_{11}^{[24]}[0,4,5,10,26,27] \oplus x_{12}^{[24]}[0,26,27] \oplus \\ & x_{13}^{[24]}[3,22,27] \oplus x_{14}^{[24]}[0,17,21,22,27] \oplus x_{15}^{[24]}[0,27]. \end{aligned}$$

Computational Result 10. *The **Linear Approximation 10** holds computationally with $\varepsilon_9 = 2^{-9.29}$. This correlation was verified using 2^{34} random samples.*

Linear Approximation 11. The following 0.25-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^{13}})$:

$$\begin{aligned} & x_0^{[24]}[8,19] \oplus x_1^{[24]}[16,27] \oplus x_2^{[24]}[0,8,11,22] \oplus \\ & x_3^{[24]}[2,3,8] \oplus x_4^{[24]}[11,27] \oplus x_5^{[24]}[10,11] \oplus \\ & x_6^{[24]}[4,5,7,8,10,18,19] \oplus x_7^{[24]}[0,13,16] \oplus x_8^{[24]}[13] \oplus \\ & x_9^{[24]}[11] \oplus x_{11}^{[24]}[0,4,5,10,26,27] \oplus x_{12}^{[24]}[0,26,27] \oplus \\ & x_{13}^{[24]}[3,22,27] \oplus x_{14}^{[24]}[0,17,21,22,27] \oplus x_{15}^{[24]}[0,27] \\ = {} & x_0^{[25]}[2,3,8,11,26,27] \oplus x_1^{[25]}[16,27] \oplus \\ & x_2^{[25]}[0,8,11,22] \oplus x_3^{[25]}[2,3,8,12,13,26] \oplus \\ & x_4^{[25]}[0,3,4,5,7,8,18,20,21,27] \oplus x_5^{[25]}[10,11] \oplus \\ & x_6^{[25]}[4,5,7,8,10,18,19] \oplus x_7^{[25]}[0,13,16] \oplus x_9^{[25]}[11] \oplus \\ & x_8^{[25]}[4,5,7,8,13,18,19,20,21,26,27] \oplus \\ & x_{11}^{[25]}[0,4,5,10,26,27] \oplus x_{12}^{[25]}[8,13,22] \oplus \\ & x_{13}^{[25]}[3,22,27] \oplus x_{14}^{[25]}[0,17,21,22,27] \oplus x_{15}^{[25]}[0,27]. \end{aligned}$$

Computational Result 11. *The **Linear Approximation 11** holds computationally with $\varepsilon_{10} = 2^{-11.67}$. This correlation was verified using 2^{36} random samples.*

By the **Piling-up Lemma**, we combine **Linear Approximations 8-11** to obtain the following 3.25-round linear approximation for Forró.

Linear Approximation 12. The following 3.25-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^{40}})$:

$$\begin{aligned} x_9^{[12]}[0] = {} & x_0^{[25]}[2,3,8,11,26,27] \oplus x_1^{[25]}[16,27] \oplus \\ & x_2^{[25]}[0,8,11,22] \oplus x_3^{[25]}[2,3,8,12,13,26] \oplus \\ & x_4^{[25]}[0,3,4,5,7,8,18,20,21,27] \oplus \\ & x_5^{[25]}[10,11] \oplus x_6^{[25]}[4,5,7,8,10,18,19] \oplus \\ & x_7^{[25]}[0,13,16] \oplus x_9^{[25]}[11] \oplus \\ & x_8^{[25]}[4,5,7,8,13,18,19,20,21,26,27] \oplus \\ & x_{11}^{[25]}[0,4,5,10,26,27] \oplus x_{12}^{[25]}[8,13,22] \oplus \\ & x_{13}^{[25]}[3,22,27] \oplus x_{14}^{[25]}[0,17,21,22,27] \oplus \\ & x_{15}^{[25]}[0,27]. \end{aligned}$$

Computational Result 12. *Using the **Piling-up Lemma** and **Computational Results 8-11**, the **Linear Approximation 12** holds computationally with $\varepsilon_{11} = \varepsilon_7\varepsilon_8\varepsilon_9\varepsilon_{10} = 2^{-(7.96+6.70+9.29+11.67)} = 2^{-35.62}$.*

We have also obtained a 3.5-round linear approximation for Forró by *MinForró*. Before introducing it, a 0.25-round linear approximation for Forró is proposed, as it will be used to calculate the computational correlation of the 3.5-round linear approximation for Forró.

Linear Approximation 13. The following 0.25-round linear

approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^{15}})$:

$$\begin{aligned}
&x_0^{[25]}[2,3,8,11,26,27] \oplus x_1^{[25]}[16,27] \oplus \\
&x_2^{[25]}[0,8,11,22] \oplus x_3^{[25]}[2,3,8,12,13,26] \oplus \\
&x_4^{[25]}[0,3,4,5,7,8,18,20,21,27] \oplus x_5^{[25]}[10,11] \oplus \\
&x_6^{[25]}[4,5,7,8,10,18,19] \oplus x_7^{[25]}[0,13,16] \oplus x_9^{[25]}[11] \oplus \\
&x_8^{[25]}[4,5,7,8,13,18,19,20,21,26,27] \oplus x_{15}^{[25]}[0,27] \oplus \\
&x_{11}^{[25]}[0,4,5,10,26,27] \oplus x_{12}^{[25]}[8,13,22] \oplus \\
&x_{13}^{[25]}[3,22,27] \oplus x_{14}^{[25]}[0,17,21,22,27] \\
=&x_0^{[26]}[3,8,10,27] \oplus x_1^{[26]}[2,3,15,16,18,19,24,29,30] \oplus \\
&x_2^{[26]}[0,8,11,22] \oplus x_3^{[26]}[2,3,8,12,13,26] \oplus \\
&x_4^{[26]}[0,3,4,5,7,8,18,20,21,27] \oplus x_5^{[26]}[8,11,21,22,26] \oplus \\
&x_6^{[26]}[4,5,7,8,10,18,19] \oplus x_7^{[26]}[0,13,16] \oplus \\
&x_8^{[26]}[4,5,7,8,12,13,18,19,20,21,26,27] \oplus \\
&x_9^{[26]}[16,21,26,27] \oplus x_{11}^{[26]}[0,4,5,10,26,27] \oplus \\
&x_{12}^{[26]}[8,13,22] \oplus x_{13}^{[26]}[6,17,21,22,29,30] \oplus \\
&x_{14}^{[26]}[0,17,21,22,27] \oplus x_{15}^{[26]}[0,27].
\end{aligned}$$

Computational Result 13. *The **Linear Approximation 13** holds computationally with $\varepsilon_{12} = 2^{-12.99}$. This correlation was verified using 2^{36} random samples.*

By the **Piling-up Lemma**, we combine **Linear Approximation 12** and **Linear Approximation 13** to the following 3.5-round linear approximation for Forró.

Linear Approximation 14. The following 3.5-round linear approximation for Forró holds with probability $\frac{1}{2}(1+\frac{1}{2^{55}})$:

$$\begin{aligned}
x_9^{[12]}[0] = &x_0^{[26]}[3,8,10,27] \oplus \\
&x_1^{[26]}[2,3,15,16,18,19,24,29,30] \oplus \\
&x_2^{[26]}[0,8,11,22] \oplus x_3^{[26]}[2,3,8,12,13,26] \oplus \\
&x_4^{[26]}[0,3,4,5,7,8,18,20,21,27] \oplus \\
&x_5^{[26]}[8,11,21,22,26] \oplus \\
&x_6^{[26]}[4,5,7,8,10,18,19] \oplus x_7^{[26]}[0,13,16] \oplus \\
&x_8^{[26]}[4,5,7,8,12,13,18,19,20,21,26,27] \oplus \\
&x_9^{[26]}[16,21,26,27] \oplus x_{11}^{[26]}[0,4,5,10,26,27] \oplus \\
&x_{12}^{[26]}[8,13,22] \oplus x_{13}^{[26]}[6,17,21,22,29,30] \oplus \\
&x_{14}^{[26]}[0,17,21,22,27] \oplus x_{15}^{[26]}[0,27].
\end{aligned}$$

Computational Result 14. *Using the **Piling-up Lemma** and **Computational Results 12 and 13**, the **Linear Approximation 14** holds computationally with $\varepsilon_{13} = \varepsilon_{11}\varepsilon_{12} = 2^{-(35.62+12.99)} = 2^{-48.61}$.*

V. Differential Trails for Forró

In this section, we will try to find good differential trails for Forró by introducing a two-stage strategy. Since the ARX-based ciphers such as Forró are designed to provide good diffusion of the differences that might appear in the internal state, the attacks that can minimize the number of active bits in the input and/or output difference are usually considered preferentially. Thus, we consider the differential trails for Forró with one active input bit, in a hope to find good differential trails for the cipher. Since the state matrix of Forró consists of 16 words in total, thus the number of possible 1-bit input differences is $16\times 32 = 512$. However, only $16\times 4 = 128$ out of all 512 1-bit input differences should be considered, as the difference can only be introduced in the four *IV* words, i.e., $x_4^{(0)}, x_5^{(0)}, x_{12}^{(0)}, x_{13}^{(0)}$. Thus, we have successfully limited the possible input differences to the 128 1-bit input differences, which reduces the search space for good differential trails efficiently. It should be noted that we also explored what happens when we consider the input differences with more active bits, e.g., two and three active bits. Unsurprisingly, the differential trails with more active bits do not perform better than the 1-bit input differences.

In order to obtain a good differential-linear distinguisher, we should consider both the differential trail and the linear approximation. Since the correlation of the differential-linear distinguisher is calculated as $\varepsilon_d \cdot \varepsilon_L^2$, it is clear to see that the influence of the linear approximation to the total correlation is squared. Thus, prioritizing the improvement of the linear approximation significantly contributes to reducing the overall complexity of the differential-linear distinguisher. In the section above, we have obtained several linear approximations with one active input bit for Forró. The active input bit is fixed at $x_{9,0}^{(3)}$, as it achieve the highest linear correlations. Since the active input bit of the linear approximation is exactly the output bit of the differential trail, we can fix the active bit of the output difference to be $x_{9,0}^{(3)}$.

Now, we have to try all 128 differential trails to find the best one among them. The 128 differential trails correspond to the 128 1-bit input differences, with the common output difference (i.e., $\Delta x_{9,0}^{(3)}$). However, the computational costs required to estimate all 128 differential trails directly are huge, as some differential trails may hold with low probabilities. To overcome this issue, we will introduce a two-stage strategy to find the best differential trail among all 128 differential trails. The process of the two-stage strategy is given as follows.

Stage 1: ***Preliminary theoretical filtering***

- Recall the description of Forró, and then we know that the four *IV* words are initialized into four initial words $x_4^{(0)}, x_5^{(0)}, x_{12}^{(0)}, x_{13}^{(0)}$. In the $Round$ function of Forró no matter odd or even, $x_4^{(0)}$ or $x_5^{(0)}$ is $x_b^{[r]}$, and $x_{12}^{(0)}$ or $x_{13}^{(0)}$ is $x_d^{[r]}$. In the SR function of Forró, $x_d^{[r]}$ participates the first modular addition operation to compute $x_{d'}^{[r]}$. Then $x_{d'}^{[r]}$ is used to calculate $x_{c'}^{[r]}$. Using the calculated $x_{c'}^{[r]}$, $x_b^{[r]}$ participates the second modular addition operation to compute $x_{b'}^{[r]}$. Thus, introducing the differential into the word $x_4^{(0)}$ or $x_5^{(0)}$ is likely to yield better differential trails than that into the word $x_{12}^{(0)}$ or $x_{13}^{(0)}$, as $\Delta x_b^{[r]}$ is able to result in less differential propagations than $\Delta x_d^{[r]}$. Now, the number of differential trails required to be estimated is reduced from 128 to 64.

TABLE IV
SUMMARY OF DIFFERENTIAL TRAILS FOR FORRÓ

Rounds	*1-bit input difference*	*1-bit output difference*	Differential correlation	Ref.
2	$\Delta x_{5,18}^{(0)}$	$\Delta x_{15,7}^{(0)}$	$2^{-8.044}$	[5]
	$\Delta x_{5,18}^{(0)}$	$\Delta x_{10,7}^{(0)}$	$2^{-8.822}$	[5]
	$\Delta x_{5,11}^{(0)}$	$\Delta x_{15,0}^{(0)}$	$2^{-9.941}$	[5]
	$\Delta x_{5,11}^{(0)}$	$\Delta x_{10,7}^{(0)}$	$2^{-10.822}$	[5]
3	$\Delta x_{5,11}^{(0)}$	$\Delta x_{9,0}^{(0)}$	$2^{-16.53}$	**This paper**
	$\Delta x_{5,25}^{(0)}$	$\Delta x_{9,0}^{(0)}$	$2^{-16.35}$	**This paper**
	$\boldsymbol{\Delta x_{5,18}^{(0)}}$	$\boldsymbol{\Delta x_{9,0}^{(0)}}$	$\mathbf{2^{-16.22}}$	**This paper**

- Recall the SR function of Forró. In the first round, the *SR* function operates on the four columns in the following order: $(x_0, x_4, x_8, x_{12}, x_3)$, $(x_1, x_5, x_9, x_{13}, x_0)$, $(x_2, x_6, x_{10}, x_{14}, x_1)$ and $(x_3, x_7, x_{11}, x_{15}, x_2)$. Because of the introduction of *Pollination*, the *SR* function is applied in a sequential order for Forró. Compared with the word $x_5^{(0)}$, the word $x_4^{(0)}$ which is updated earlier than $x_5^{(0)}$ has higher diffusion. Thus, it is a natural consequence that introducing the differential into the word $x_5^{(0)}$ is likely to yield better differential trails than that into the word $x_4^{(0)}$. Now, the number of differential trails required to be estimated is further reduced from 64 to 32.

Stage 2: ***Exhaustive detection***

- In this stage, we focus on trying the differential trails obtained in the first stage above. Since most of the differential trails (i.e., 96 out of 128) have been filtered out in the first stage, we can exhaust the remaining 32 differential trails to find the best one among them. For each of 32 differential trails, we estimate its differential correlation with 2^6 random keys and 2^{36} random *IVs*. The estimation process has been done within 3.67 days on a common PC.

By using the two-stage strategy above, we have filtered out 96 differential trails in the first stage through theoretical analysis of Forró, and then successfully found the best differential trails among the remaining 32 differential trails by making exhaustive detection in the second stage. The obtained results are summarized in Table IV. As shown in this table, the best 3-round differential trail we obtained is $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$, which achieves the highest differential correlation of $\varepsilon_d = 2^{-16.22}$.

Comparisons with the existing results. In the specification of Forró [3] (and the extended version [5]), the designers of Forró tested all combinations of single-bit input difference and single-bit output difference, and obtained four 2-round differential trails, as shown in Table IV. In this paper, we have introduced a two-stage strategy to find good differential trails for Forró. The advantage of this new strategy is that it can filter out the differential trails with low differential correlations in the first stage, which significantly reduces the difficulty of estimating differential correlations. This is the key reason that we have succeeded in finding several 3-round differential trails for Forró, while only 2-round differential trails for Forró are found in [3] and [5].

VI. DIFFERENTIAL-LINEAR DISTINGUISHERS FOR FORRÓ

In this section, we use the 3-round differential trail with correlation $\varepsilon_d = 2^{-16.22}$ and the linear approximations obtained above to construct differential-linear distinguishers for Forró.

Differential-Linear Distinguisher for 4-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 1-round linear trail **Linear Approximation 1**. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_0 = 1$ of **Computational Result 1**, we get a differential-linear distinguisher for 4-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_0^{-4} = 2^{32.44}$.

Differential-Linear Distinguisher for 5-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 2-round linear trail **Linear Approximation 2**. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_1 = 2^{-3.39}$ of **Computational Result 2**, we get a differential-linear distinguisher for 5-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_1^{-4} = 2^{46}$.

Experimental Verifications of the Distinguishers for 4- and 5-Round Forró. Since the complexities of the differential-linear distinguishers for 4- and 5-round Forró are within the calculation capability of a common PC, we have implemented these distinguishers to demonstrate their correctness. For 4-round Forró, the experiment utilizes a sample size of 2^{34} and requires a total running time of 2.38 minutes on a common PC with an Intel(R) Core(TM) i5-14400 processor (2.50 GHz) and an NVIDIA GeForce RTX 4060 GPU. For 5-round Forró, the experiment utilizes a sample size of 2^{48} and requires a total running time of 9.58 days on the same common PC. As results, the 4- and 5-round Forró have been successfully distinguished from a pseudorandom permutation. Thus, the experimental results confirm the correctness of these distinguishers.

Differential-Linear Distinguisher for 5.25-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 2.25-round linear trail **Linear**

Approximation 3. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_2 = 2^{-4.39}$ of **Computational Result 3**, we get a differential-linear distinguisher for 5.25-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_2^{-4} = 2^{50}$.

Differential-Linear Distinguisher for 5.5-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 2.5-round linear trail **Linear Approximation 4**. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_3 = 2^{-7.97}$ of **Computational Result 4**, we get a differential-linear distinguisher for 5.5-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_3^{-4} = 2^{64.32}$.

Differential-Linear Distinguisher for 5.75-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 2.75-round linear trail **Linear Approximation 5**. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_4 = 2^{-13.67}$ of **Computational Result 5**, we get a differential-linear distinguisher for 5.75-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_4^{-4} = 2^{87.12}$.

Differential-Linear Distinguisher for 6-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 3-round linear trail **Linear Approximation 7**. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_6 = 2^{-21.37}$ of **Computational Result 7**, we get a differential-linear distinguisher for 6-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_6^{-4} = 2^{117.92}$.

Differential-Linear Distinguisher for 6.25-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 3.25-round linear trail **Linear Approximation 12**. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_{11} = 2^{-35.62}$ of **Computational Result 12**, we get a differential-linear distinguisher for 6.25-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_{11}^{-4} = 2^{174.92}$.

Differential-Linear Distinguisher for 6.5-Round Forró. The distinguisher consists of the 3-round differential trail $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$ and the 3.5-round linear trail **Linear Approximation 14**. Considering the differential correlation $\varepsilon_d = 2^{-16.22}$ of the 3-round differential trail and the experimental linear correlation $\varepsilon_{13} = 2^{-48.61}$ of **Computational Result 14**, we get a differential-linear distinguisher for 6.5-round Forró with complexity $\varepsilon_d^{-2} \cdot \varepsilon_{13}^{-4} = 2^{226.88}$.

VII. PNB-based Key Recovery Attacks on Forró

When combined with the PNB technique, the differential-linear distinguishers can be converted to key recovery attacks. In this section, we will present the PNB-based key recovery attacks on Forró.

In Section V, we have obtained a 3-round differential trail with differential correlation $\varepsilon_d = 2^{-16.22}$, i.e., $\Delta x_{5,18}^{(0)} \rightarrow \Delta x_{9,0}^{(3)}$. Recall **Linear Approximation 1**, we know that the 1-round linear trail $x_{9,0}^{(3)} = x_2^{(4)}[0] \oplus x_9^{(4)}[0] \oplus x_{14}^{(4)}[0] \oplus x_{14}^{(4)}[27]$ holds with probability 1. By combing the 3-round differential trail and 1-round linear approximation, we can obtain the 4-round differential-linear distinguisher $\Delta x_{5,18}^{(0)} \rightarrow \Gamma^{(4)}$ with correlation $\varepsilon_d' = \varepsilon_d \cdot \varepsilon_0^2 = 2^{-16.22}$, where $\Gamma^{(4)} = x_2^{(4)}[0] \oplus x_9^{(4)}[0] \oplus x_{14}^{(4)}[0] \oplus x_{14}^{(4)}[27]$. The 4-round differential-linear distinguisher will be used in the following key recovery attacks on Forró.

A. PNB-based Key Recovery Attack on 5.5-Round Forró

The PNB-based key recovery attack on 5.5-round Forró consists of 4 rounds in the forward direction and 1.5 rounds in the backward direction. In the backward direction, we utilize the three-step strategy to find the PNBs. The process of finding the PNBs is described as follows.

At first, we assign a threshold $\gamma_{prelim} = 0.7$ to shortlist the PNBs, and then a total of 201 PNBs are obtained. By assigning the threshold $\gamma_{direct} = 0.996$, we directly include 148 out of 201 PNBs into the PNB set. The list of the obtained 148 PNBs is given as follows. Note that the second and third steps are not used here, since the number of PNBs is already enough for the attack.

{ 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 35, 36, 37, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 234, 235 }

By assigning $100\cdots00$ to the consecutive PNBs and assigning 0 to the PNBs that are not consecutive, we obtain a backward correlation $\varepsilon_a = 2^{-0.0056}$ using 2^{20} random samples. For $\alpha = 114$, the required number of samples can be computed as

$$N = \left(\frac{\sqrt{\alpha \log 4} + 3\sqrt{1-\varepsilon_d'^2 \varepsilon_a^2}}{\varepsilon_d' \varepsilon_a} \right)^2 \approx 2^{40.37}.$$

Thus, the PNB-based key recovery attack on 5.5-round Forró has a time complexity of $2^{149.20}$, with a data complexity of $2^{40.37}$.

B. Key Recovery Attack on 6-Round Forró

The PNB-based key recovery attack on 6-round Forró consists of 4 rounds in the forward direction and 2 rounds in the backward direction. In the backward direction, we utilize the three-step strategy to find the PNBs. Now we present the first-ever key recovery attack on 6-round Forró, which is described as follows.

At first, we assign a threshold $\gamma_{prelim} = 0.3$ to shortlist the PNBs, and then a total of 174 PNBs are obtained. By assigning the threshold $\gamma_{direct} = 0.8$, we directly include 124

out of 174 PNBs into the PNB set. The list of the obtained 124 PNBs is given as follows.

{ 0, 1, 2, 3, 8, 9, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 35, 36, 37, 38, 39, 40, 41, 59, 75, 76, 77, 78, 79, 115, 116, 117, 118, 119, 131, 132, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 159, 160, 161, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 229, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 251, 252, 253, 254 }

We use the second step to add more PNBs into the PNB set. In each iteration, we assign random values to the obtained 124 PNBs. Among the remaining $(174 - 124 = 50)$ key bits, we change the value one at a time and estimate its backward bias. The key bit with the highest bias is selected into the PNB set. Finally, we include 27 PNBs into the PNB set. The list of the obtained 151 PNBs is given as follows. Note that the third step is not used here, since the number of PNBs is already enough for the attack.

{ 0, 1, 2, 3, 8, 9, 10, 11, 12, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 35, 36, 37, 38, 39, 40, 41, 42, 48, 49, 50, 59, 60, 61, 62, 63, 67, 75, 76, 77, 78, 79, 80, 107, 115, 116, 117, 118, 119, 120, 131, 132, 133, 134, 135, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 229, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 251, 252, 253, 254, 255 }

By assigning $100\cdots00$ to the consecutive PNBs and assigning 0 to the PNBs that are not consecutive, we obtain a backward correlation $\varepsilon_a = 2^{-2.66}$ using 2^{20} random samples. For $\alpha = 112$, the required number of samples can be computed as

$$N = \left(\frac{\sqrt{\alpha \log 4}+3\sqrt{1-\varepsilon_d'^2\varepsilon_a^2}}{\varepsilon_d'\varepsilon_a}\right)^2 \approx 2^{45.65}.$$

Thus, the PNB-based key recovery attack on 6-round Forró has a time complexity of $2^{151.84}$, with a data complexity of $2^{45.65}$.

C. Key Recovery Attack on 6.5-round Forró

The PNB-based key recovery attack on 6.5-round Forró consists of 4 rounds in the forward direction and 2.5 rounds in the backward direction. In the backward direction, we utilize the three-step strategy to find the PNBs. Now we present the first-ever key recovery attack on 6.5-round Forró, which is described as follows.

At first, we assign a threshold $\gamma_{prelim} = 0.15$ to shortlist the PNBs, and then a total of 107 PNBs are obtained. By assigning the threshold $\gamma_{direct} = 0.3$, we directly include 85 out of 107 PNBs into the PNB set. The list of the obtained 85 PNBs is given as follows.

{ 27, 28, 29, 30, 35, 36, 67, 68, 69, 75, 76, 77, 78, 79, 80, 81, 82, 91, 115, 116, 117, 118, 163, 164, 165, 171, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 229, 230, 231, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 251, 252, 253, 254, 255 }

We use the second step to add more PNBs into the PNB set. In each iteration, we assign random values to the obtained 85 PNBs. Among the remaining $(107-85 = 22)$ key bits, we change the value one at a time and estimate its backward bias. The key bit with the highest bias is selected into the PNB set. As a result, we include 10 PNBs into the PNB set in this step.

In the third step, we assign random values to the obtained 95 PNBs and randomly change one of the $(107-85-10 = 12)$ candidates. We change the value of one bit at a time and arrange them in descending order of their backward biases. We directly select the top 5 bits among them to form the final PNB set that consists of $95 + 5 = 100$ bits. The list of the obtained 100 PNBs is given as follows.

{ 27, 28, 29, 30, 31, 35, 36, 37, 67, 68, 69, 70, 75, 76, 77, 78, 79, 80, 81, 82, 91, 92, 93, 99, 100, 115, 116, 117, 118, 119, 141, 155, 163, 164, 165, 166, 171, 176, 181, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 229, 230, 231, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 251, 252, 253, 254, 255 }

By assigning $100\cdots00$ to the consecutive PNBs and assigning 0 to the PNBs that are not consecutive, we obtain a backward correlation $\varepsilon_a = 2^{-9.03}$ using 2^{28} random samples. For $\alpha = 48$, the required number of samples can be computed as

$$N = \left(\frac{\sqrt{\alpha \log 4}+3\sqrt{1-\varepsilon_d'^2\varepsilon_a^2}}{\varepsilon_d'\varepsilon_a}\right)^2 \approx 2^{57.45}.$$

Thus, the PNB-based key recovery attack 6.5-round Forró has a time complexity of $2^{213.49}$, with a data complexity of $2^{57.45}$.

D. Key Recovery Attack on 6.75-round Forró

The PNB-based key recovery attack on 6.75-round Forró consists of 4 rounds in the forward direction and 2.75 rounds in the backward direction. In the backward direction, we utilize the three-step strategy to find the PNBs. Now we present the first-ever key recovery attack on 6.75-round Forró, which is described as follows.

At first, we assign a threshold $\gamma_{prelim} = 0.1$ to shortlist the PNBs, and then a total of 70 PNBs are obtained. By assigning the threshold $\gamma_{direct} = 0.2$, we directly include 52 out of 70 PNBs into the PNB set. The list of the obtained 52 PNBs is given as follows.

$\{$ 27, 28, 29, 30, 31, 35, 36, 37, 75, 76, 99, 115, 116, 117, 118, 119, 163, 164, 165, 171, 172, 176, 187, 188, 189, 190, 191, 224, 225, 229, 230, 231, 232, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 251, 252, 253, 254, 255 $\}$

We use the second step to add more PNBs into the PNB set. In each iteration, we assign random values to the obtained 52 PNBs. Among the remaining $(70 - 52 = 18)$ key bits, we change the value one at a time and estimate its backward bias. The key bit with the highest bias is selected into the PNB set. As a result, we include 10 PNBs into the PNB set in this step.

In the third step, we assign random values to the obtained 62 PNBs and randomly change one of the $(70 - 52 - 10 = 8)$ candidates. We change the value of one bit at a time and arrange them in descending order of their backward biases. We directly select the top 4 bits among them to form the final PNB set that consists of $62 + 4 = 66$ bits. The list of the obtained 66 PNBs is given as follows.

$\{$ 11, 27, 28, 29, 30, 31, 35, 36, 37, 38, 75, 76, 77, 99, 100, 101, 115, 116, 117, 118, 119, 120, 141, 155, 163, 164, 165, 166, 171, 172, 176, 177, 181, 187, 188, 189, 190, 191, 203, 221, 224, 225, 226, 229, 230, 231, 232, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 251, 252, 253, 254, 255 $\}$

By assigning $100\cdots00$ to the consecutive PNBs and assigning 0 to the PNBs that are not consecutive, we obtain a backward correlation $\varepsilon_a = 2^{-12.06}$ using 2^{28} random samples. For $\alpha = 8$, the required number of samples can be computed as

$$N = \left(\frac{\sqrt{\alpha \log 4} + 3\sqrt{1-\varepsilon_d'^2 \varepsilon_a^2}}{\varepsilon_d' \varepsilon_a} \right)^2 \approx 2^{61.87}.$$

Thus, the PNB-based key recovery attack 6.75-round Forró has a time complexity of $2^{251.97}$, with a data complexity of $2^{61.87}$.

VIII. Conclusion

At ASIACRYPT 2022, Coutinho et al. [3] proposed a new ARX-based stream cipher called Forró based on the introduced design strategy *Pollination*. This paper demonstrates the first comprehensive security analysis of the Forró stream cipher through the full automated tool. We propose a fully automated MILP model to derive linear approximations for the Forró stream cipher, called *MinForró*, and a two-stage strategy to search for single-bit differential trails with high differential correlations. As applications, several differential-linear distinguishers and key recovery attacks on reduced-round Forró are presented, which achieve significant advancements over the existing records of attacked rounds. Specifically, we present the first differential-linear distinguisher for Forró that reaches 6.5 rounds, which is a significant advancement over the existing record of 5.5 rounds. We also present the first key recovery attack on Forró that reaches 6.75 rounds, which is a significant advancement over the existing record of 5.5 rounds. These cryptanalytic results definitely demonstrates the superiority of the cryptanalytic techniques proposed in this paper.

The methodological approach introduced in this paper exhibits a significant generalizability. The constrained optimization approach underlying *MinForró* can be readily adapted to other ARX-based ciphers, particularly those with sequential *SubRound* structures. While reviewing the design principles of its components, we noted that the constants are chosen solely for the cultural reasons. This has raised certain concerns, although the designers asserted that this choice does not compromise the cipher's security or performance. We expect that our work has the potential to be further improved.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China under Grant number 62472439.

References

[1] D. J. Bernstein, "The Salsa20 Family of Stream Ciphers," in *New Stream Cipher Designs*, Berlin, Heidelberg: Springer, 2008, vol. 4986, pp. 84–97.

[2] D. J. Bernstein, "ChaCha, a variant of Salsa20," In *Proc. Workshop Rec. SASC 2008*, Lausanne, Switzerland, Feb. 2008, vol. 8, pp. 3–5.

[3] M. Coutinho et al., "Latin dances reloaded: Improved cryptanalysis against salsa and chacha, and the proposal of forró," in *Proc. ASIACRYPT 2022*. Taipei, Taiwan, Dec. 2022, vol. 13791, pp. 256–286.

[4] M. Coutinho, I. Passos, and F. Borges,"The design and implementation of xforró14-poly1305: a new authenticated encryption scheme," in *Proc. SBSeg 2023*. Sep. 2023, pp. 456–469. Available: https://doi.org/10.5753/sbseg.2023.232879

[5] M. Coutinho, I. Passos, J. C. G. Vásquez, S. Sarkar, F. L. L. de Mendonça, R. T. de Sousa, and F. Borges, "Latin dances reloaded: Improved cryptanalysis against salsa and chacha, and the proposal of forró," *Journal of Cryptology*, vol. 36, no. 3, pp. 1-57, 2023.

[6] N. Sharma, H. Garai, and S. Dey, "Breaching forró's security with differential-linear foray," *IEEE Access*, vol. 12, pp. 99175–99182, 2024.

[7] X. Gong, Q. Wang, Y. Hao, L. Jiao, and X. Hu, "Persistence of hourglass (-like) structure: improved differential-linear distinguishers for several arx ciphers," in *Proc. ASIACRYPT 2025*. Melbourne, Australia, Dec. 2025, vol. 16245, pp. 317–351.

[8] S. K. Langford and M. E. Hellman, "Differential-linear cryptanalysis," in *Proc. CRYPTO 1994*, Santa Barbara, California, USA, Aug. 1994, vol. 839, pp. 17–25.

[9] J. P. Aumasson, S. Fischer, S. Khazaei, W. Meier, and C. Rechberger, "New features of latin dances: Analysis of salsa, ChaCha, and rumba," in *Proc. FSE 2008*, Lausanne, Switzerland, Feb. 2008, vol. 5086, pp. 470–488.

[10] S. Dey, H. K. Garai, S. Sarkar, and N. K. Sharma, "Revamped differential-linear cryptanalysis on reduced round ChaCha," in *Proc. EUROCRYPT 2022*, Trondheim, Norway, May. 2022, vol. 13277, pp. 86–114.

[11] Z. Shi, B. Zhang, D. Feng, and W. Wu, "Improved key recovery attacks on reduced-round salsa20 and chacha," in *Proc. ICISC 2012*, Seoul, Korea, Nov. 2012, vol. 7839, pp. 337–351.

[12] S. Maitra, "Chosen IV cryptanalysis on reduced round ChaCha and salsa," *Discrete Applied Mathematics*, vol. 208, pp. 88–97, 2016.

[13] A. R. Choudhuri and S. Maitra, "Significantly improved multi-bit differentials for reduced round salsa and ChaCha," *IACR Trans. Symmetric Cryptol*, vol. 2016, no. 2, pp. 261–287, 2017.

[14] S. Dey and S. Sarkar, "Improved analysis for reduced round salsa and chacha," *Discrete Applied Mathematics*, vol. 227, pp. 58–69, 2017.

[15] M. Coutinho and T. C. S. Neto, "New multi-bit differentials to improve attacks against chacha," *Cryptol. ePrint Arch.*, Tech. Rep. 2020/350. [Online]. Available: https://eprint.iacr.org/2020/350

[16] C. Beierle, M. Broll, F. Canale, N. David, A. Flórez-Gutiérrez, G. Leander, M. Naya-Plasencia, and Y. Todo, "Improved differential-linear attacks with applications to arx ciphers," *Journal of Cryptology*, vol. 35, no. 4, pp. 1-62, 2022.

[17] M. Coutinho and T. C. Souza Neto, "Improved linear approximations to arx ciphers and attacks against chacha," in *Proc. EUROCRYPT 2021*. Zagreb, Croatia, Oct. 2021, vol. 12696, pp. 711–740.
[18] S. Miyashita, R. Ito, and A. Miyaji, "PNB-focused differential cryptanalysis of ChaCha stream cipher," in *Proc. ACISP 2022*. Wollongong, NSW, Australia, Nov. 2022, vol. 13494, pp. 46–66.
[19] S. Dey, H. K. Garai, S. Sarkar, and N. K. Sharma, "Enhanced differential-linear attacks on reduced round chacha," *IEEE Trans. Inf. Theory*, vol. 69, no. 8, pp. 5318–5336, 2023.
[20] S. Wang, M. Liu, S. Hou, and D. Lin, "Moving a step of ChaCha in syncopated rhythm," in *Proc. CRYPTO 2023*. Santa Barbara, USA, Aug. 2023, vol. 14083, pp. 273–304.
[21] E. Bellini, D. Gerault, J. Grados, R. H. Makarim, and T. Peyrin, "Boosting differential-linear cryptanalysis of chacha7 with milp," *IACR Trans. Symmetric Cryptol*, vol. 2023, no. 2, pp. 189–223, 2023.
[22] S. Dey, H. K. Garai, and S. Maitra, "Cryptanalysis of reduced round ChaCha – new attack and deeper analysis," *IACR Trans. Symmetric Cryptol*, vol. 2023, no. 1, pp. 89–110, 2023.
[23] Z. Xu, H. Xu, L. Tan, and W. Qi, "Differential-linear cryptanalysis of reduced round chacha," *IACR Trans. Symmetric Cryptol*, vol. 2024, no. 2, pp. 166–189, 2024.
[24] S. Dey, "Advancing the idea of probabilistic neutral bits: First key recovery attack on 7.5 round chacha," *IEEE Trans. Inf. Theory*, vol. 70, no. 8, pp. 6091–6106, 2024.
[25] N. K. Sharma, S. Dey, S. Sarkar, and S. Maitra, "On improved cryptanalytic results against ChaCha for reduced rounds $\geq$ 7," in *Proc. INDOCRYPT 2024*, Odisha, India, Dec. 2024, vol. 15496, pp. 29–52.
[26] H. K. Garai and S. Dey, "A multi-step key recovery attack on reduced round salsa and chacha," *Cryptologia*, vol. 49, no. 3, pp. 252–267, 2025.
[27] K. Fu, M. Wang, Y. Guo, S. Sun, and L. Hu, "Milp-based automatic search algorithms for differential and linear trails for speck," in *Proc. FSE 2016*, Bochum, Germany, Mar. 2016, vol. 9783, pp. 268–288.